Additional Task

Name:Shaik Khaja Basha
Batch : Batch 11
Date : 26.07.2025
Task : terraform with jenkins

Task:

Create two files and two directors using terraform with help of jenkins pipeline

Create a Jenkins pipeline that provisions infrastructure using Terraform. The pipeline should:

1. **Automatically perform terraform init, validate, plan, and apply stages.**
2. **All Terraform files (main.tf, variables.tf, terraform.tfvars, etc.) should be stored in a GitHub repository (e.g., terraform-jenkins-pipeline).**
3. **Clone this repository in your Jenkins pipeline.**
4. **Design the Jenkinsfile to execute the Terraform steps sequentially.**
5. **The files should be stored in the Jenkins workspace (agent) during execution.**
6. **After running the pipeline, verify that the Terraform resources are created successfully two files and two directories in your local or target environment.**
7. **Take screenshots of the output/results as proof**

Jenkins Pipeline for Provisioning Files and Directories with Terraform

Below is a step-by-step guide to create two files and two directories using Terraform, orchestrated via a Jenkins Pipeline. This approach integrates version control (GitHub), automated infrastructure provisioning (Terraform), and CI/CD (Jenkins).

1. Prepare GitHub Repository

Store all Terraform configuration files (main.tf, variables.tf, terraform.tfvars, etc.) in a GitHub repository, such as Jenkins-terraform-pipeline/.

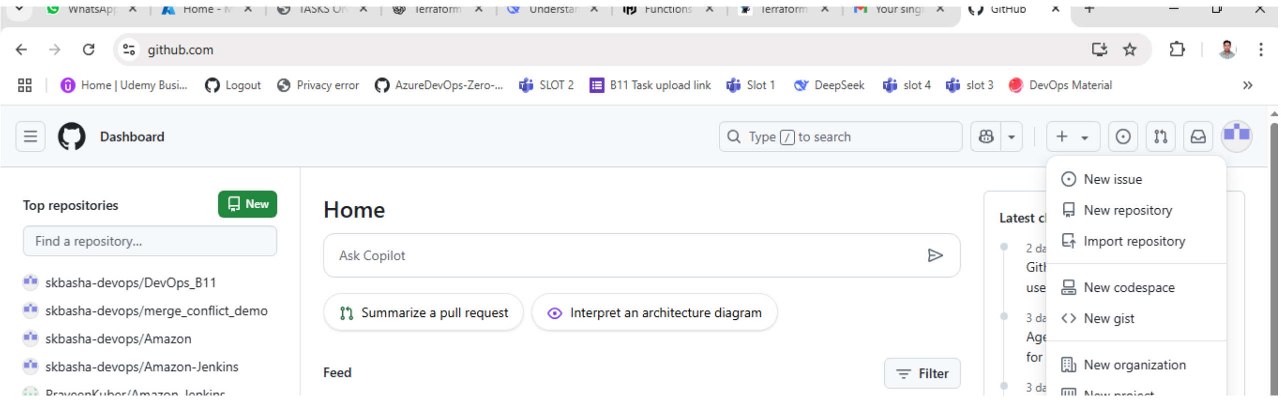
Example file structure:

Jenkins-terraform-pipeline/

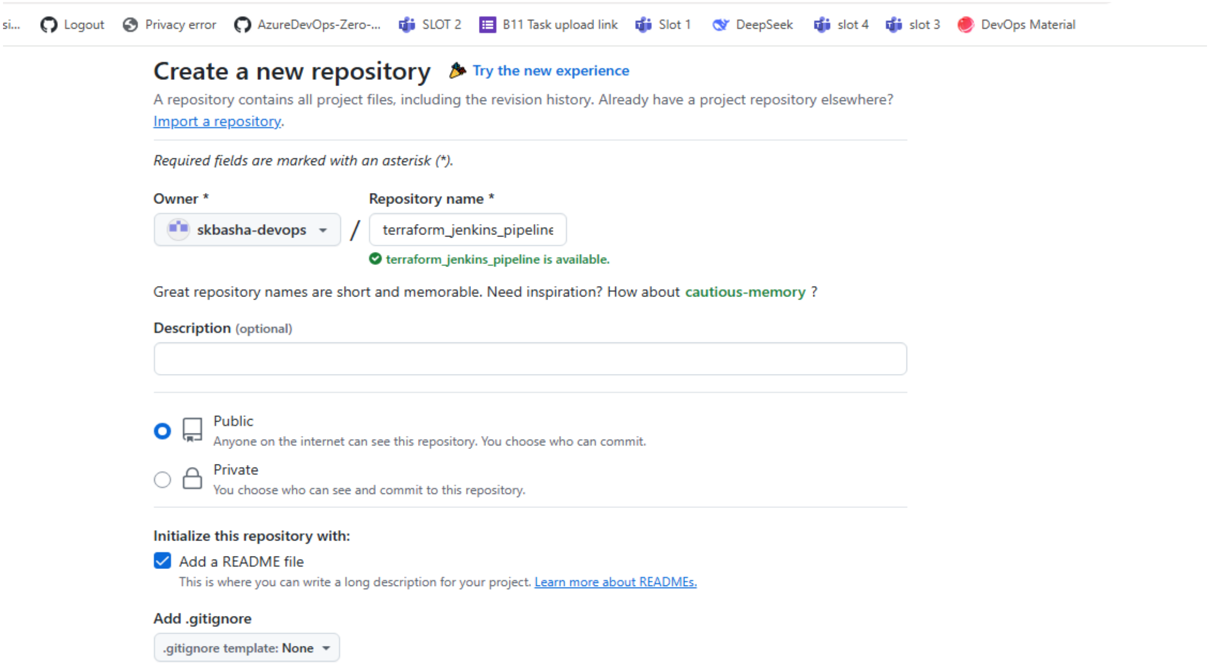
terraform-jenkins-pipeline/

```
|--main.tf
|--Jenkinsfile
|-- outputs.tf (optional)
|-- .gitignore (optional)
└── README.md (optional)
```

Create a new repository by clicking on New repository



Enter the details as follows



Create a new repository

Try the new experience

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * skbasha-devops / Repository name * terraform_jenkins_pipeline

✓ terraform_jenkins_pipeline is available.

Great repository names are short and memorable. Need inspiration? How about cautious-memory ?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

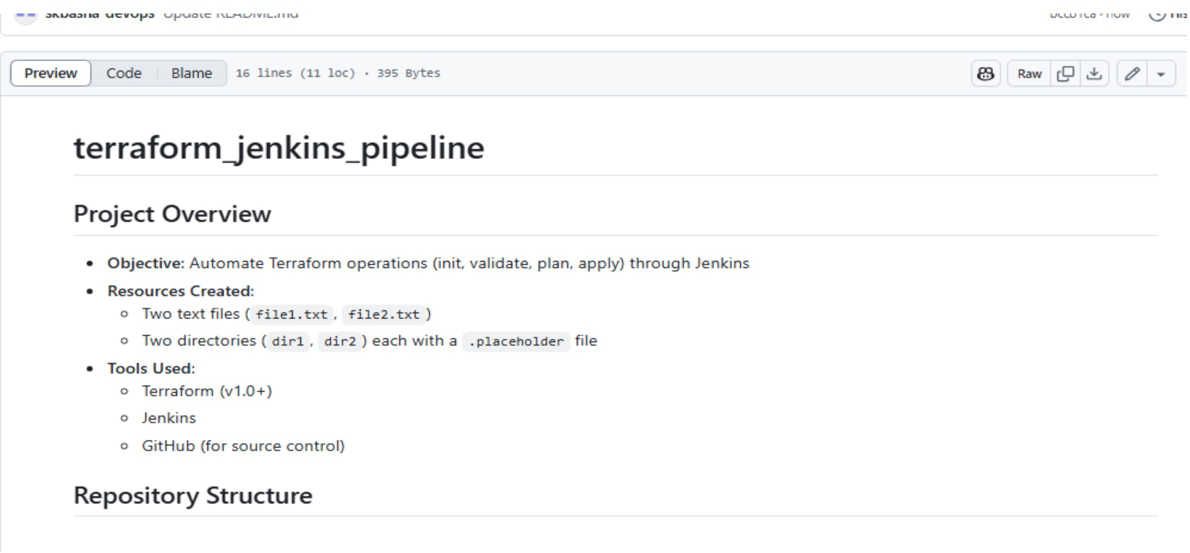
Initialize this repository with:

☑ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Enter the details for README.md file


skbasha-devops Update README.md
Preview
Code
Blame
16 lines (11 loc) · 395 Bytes
Raw

terraform_jenkins_pipeline

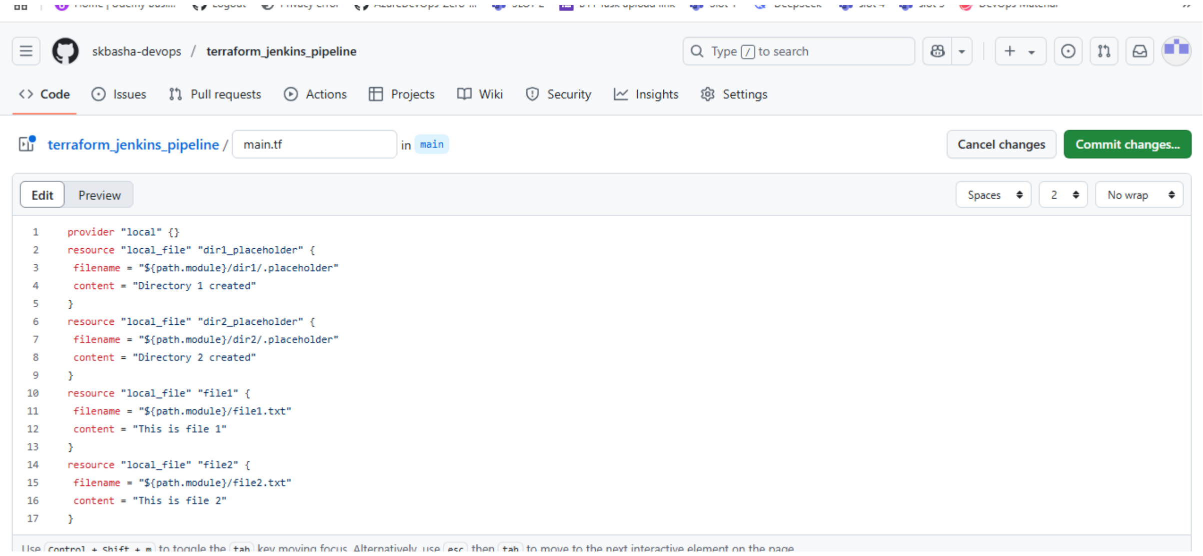
Project Overview

- **Objective:** Automate Terraform operations (init, validate, plan, apply) through Jenkins
- **Resources Created:**
  - Two text files (`file1.txt` , `file2.txt`)
  - Two directories (`dir1` , `dir2`) each with a `.placeholder` file
- **Tools Used:**
  - Terraform (v1.0+)
  - Jenkins
  - GitHub (for source control)

Repository Structure

Create a new file and enter the name main.tf

Add the contents to file and click on Commit changes



skbasha-devops / terraform_jenkins_pipeline

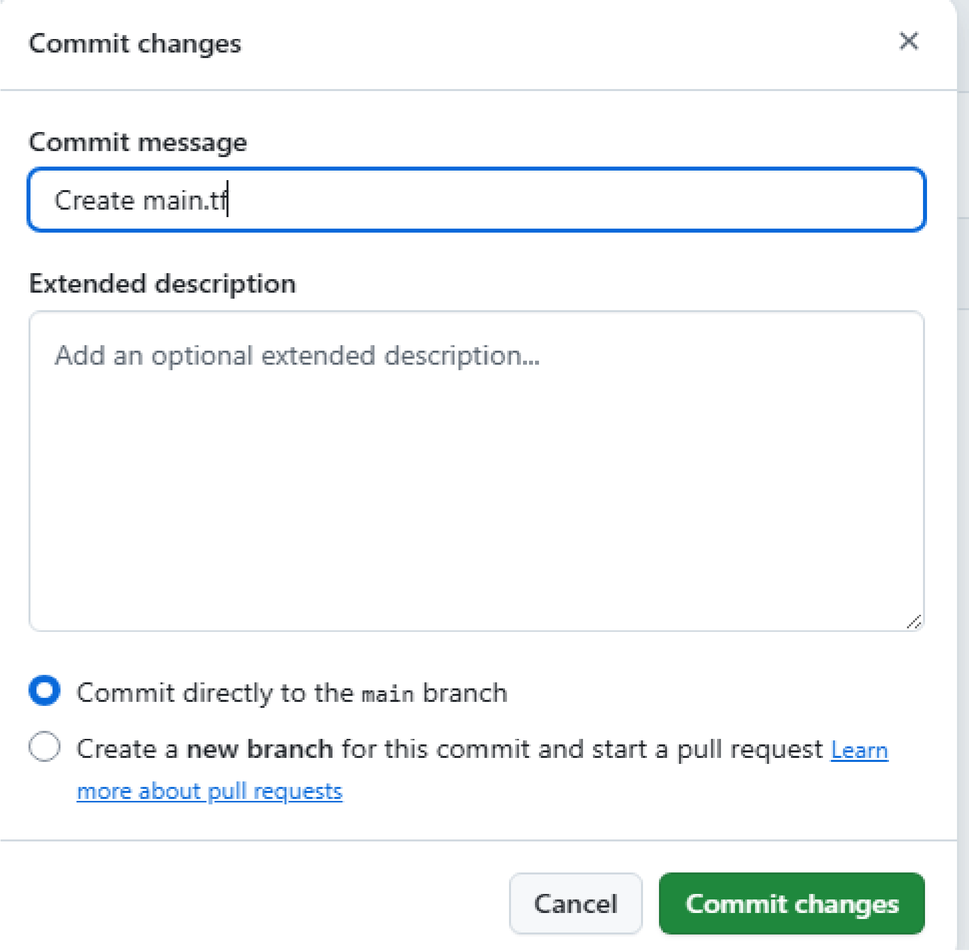
terraform_jenkins_pipeline / main.tf in main

```
provider "local" {}
resource "local_file" "dir1_placeholder" {
 filename = "${path.module}/dir1/.placeholder"
 content = "Directory 1 created"
}
resource "local_file" "dir2_placeholder" {
 filename = "${path.module}/dir2/.placeholder"
 content = "Directory 2 created"
}
resource "local_file" "file1" {
 filename = "${path.module}/file1.txt"
 content = "This is file 1"
}
resource "local_file" "file2" {
 filename = "${path.module}/file2.txt"
 content = "This is file 2"
}
```

Use control + shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page

If we use ${path.module} in the root module, it resolves to . (the current directory).

Enter the Commit message and click on Commit changes



Commit changes

Commit message

Create main.tf

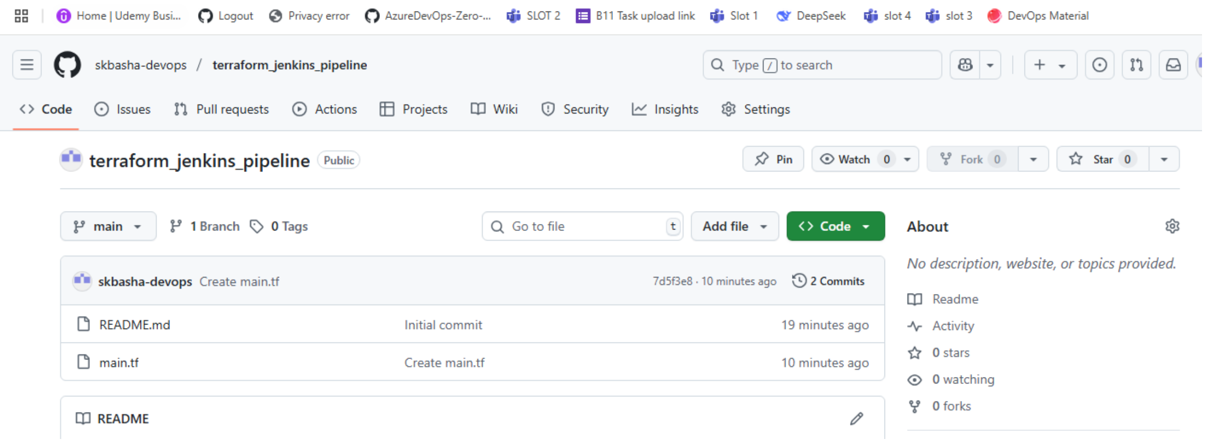
Extended description

Add an optional extended description...

● Commit directly to the `main` branch

○ Create a **new branch** for this commit and start a pull request Learn more about pull requests

main.tf will be added in main branch

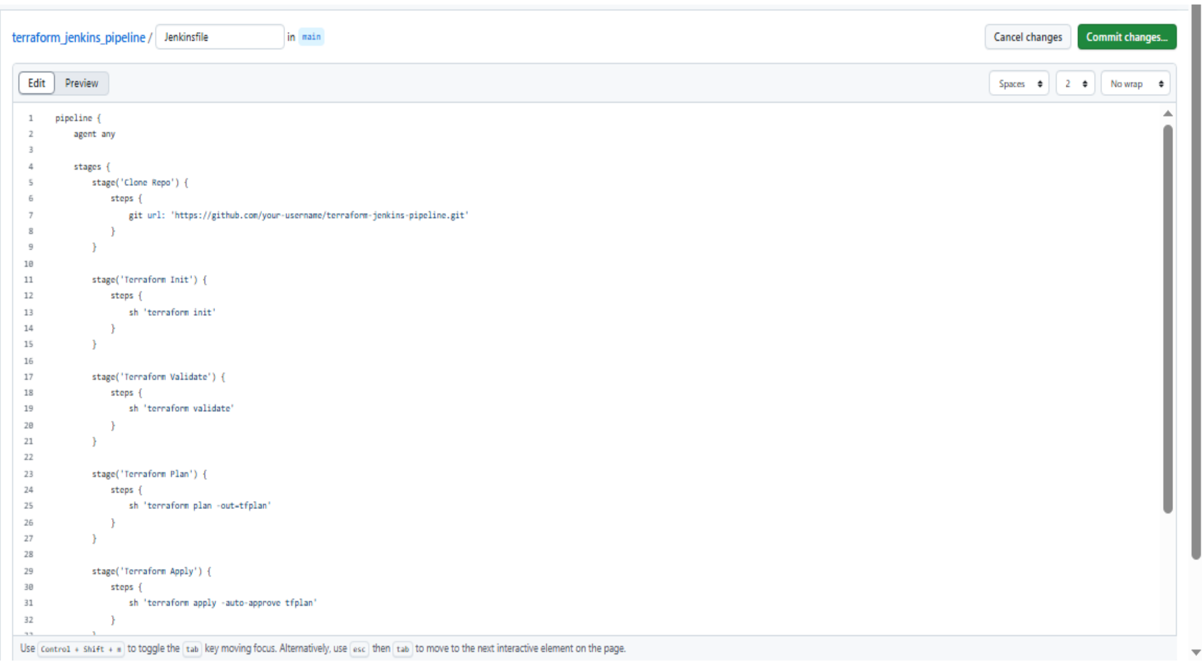


No description, website, or topics provided.

skbasha-devops Create main.tf 7d5f3e8 · 10 minutes ago 2 Commits

README.md	Initial commit	19 minutes ago
main.tf	Create main.tf	10 minutes ago

Now we need to add the Jenkinsfile, enter the details as follows and click on Commit changes

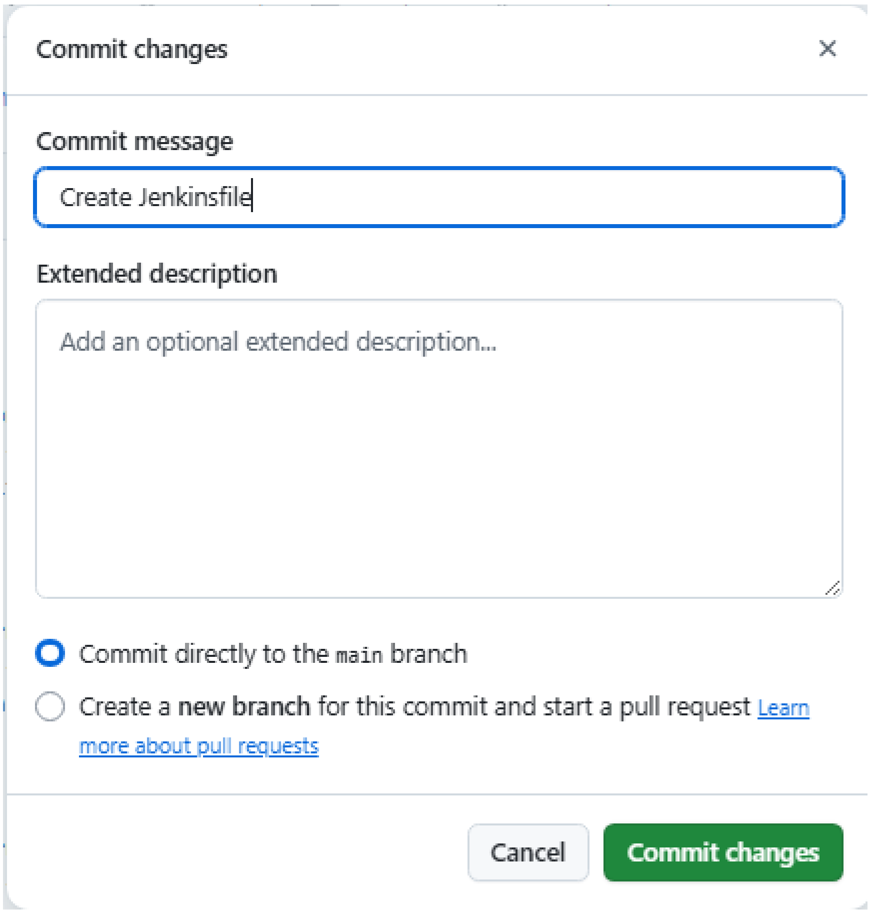


Jenkinsfile

```
pipeline {
    agent any

    stages {
        stage('Clone Repo') {
            steps {
                git url: 'https://github.com/your-username/terraform-jenkins-pipeline.git'
            }
        }

        stage('Terraform Init') {
            steps {
                sh 'terraform init'
            }
        }

        stage('Terraform Validate') {
            steps {
                sh 'terraform validate'
            }
        }

        stage('Terraform Plan') {
            steps {
                sh 'terraform plan -out=tfplan'
            }
        }

        stage('Terraform Apply') {
            steps {
                sh 'terraform apply -auto-approve tfplan'
            }
```

Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.

Enter the Commit message and click on Commit changes



Commit changes

Commit message

Create Jenkinsfile

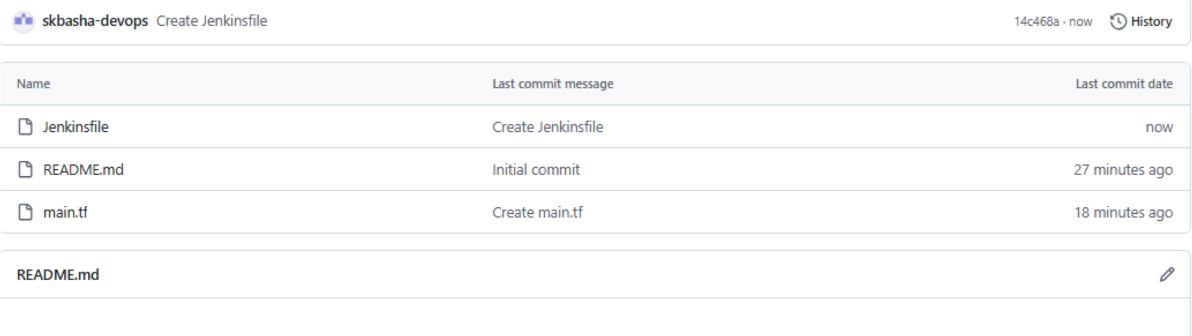
Extended description

Add an optional extended description...

● Commit directly to the main branch

○ Create a **new branch** for this commit and start a pull request Learn more about pull requests

Jenkinsfile is added

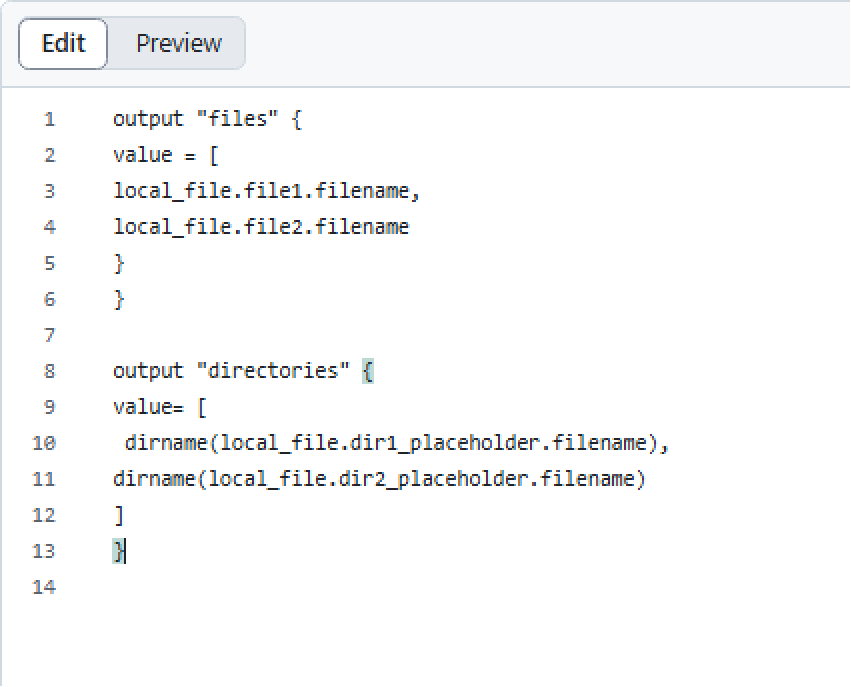


skbasha-devops Create Jenkinsfile 14c468a · now History

Name	Last commit message	Last commit date
Jenkinsfile	Create Jenkinsfile	now
README.md	Initial commit	27 minutes ago
main.tf	Create main.tf	18 minutes ago

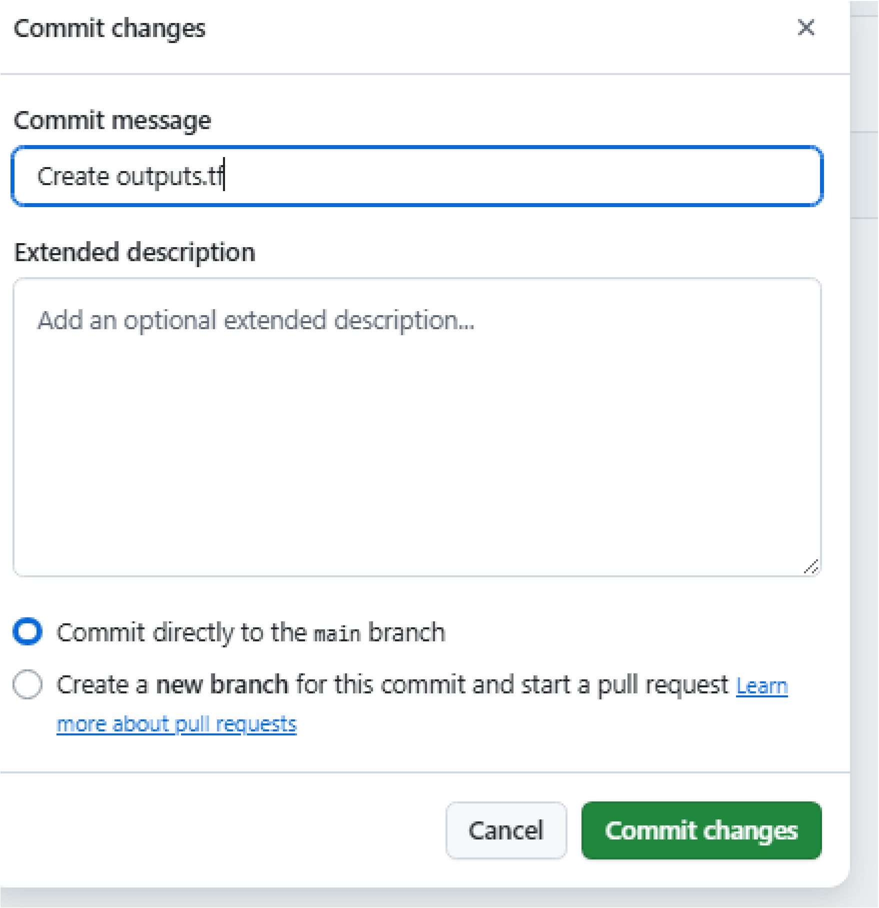
README.md

Add the outputs.tf file enter the details as follows and click on Commit changes



```
output "files" {
value = [
local_file.file1.filename,
local_file.file2.filename
}
}

output "directories" {
value= [
 dirname(local_file.dir1_placeholder.filename),
dirname(local_file.dir2_placeholder.filename)
]
}
```

Enter the commit message and click on Commit changes


Commit changes

Commit message

Create outputs.tf

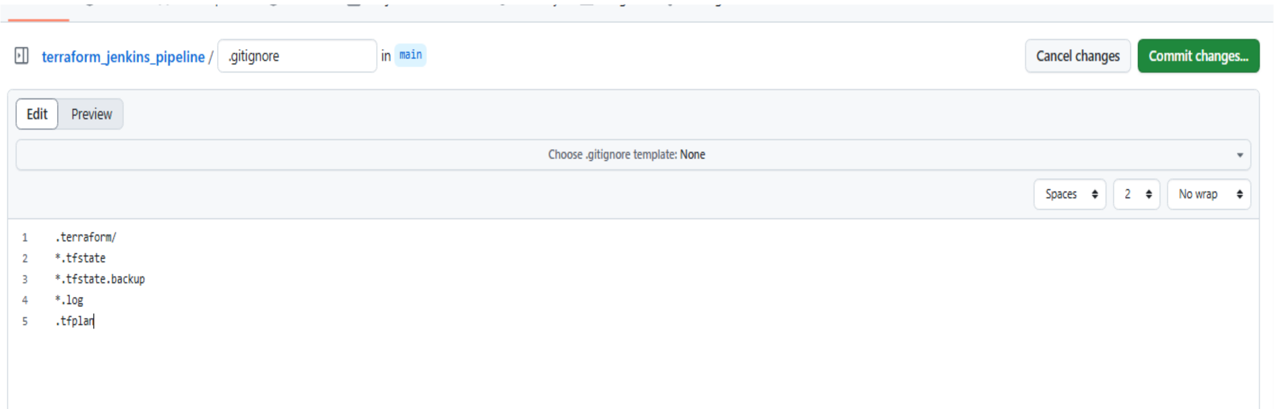
Extended description

Add an optional extended description...

● Commit directly to the `main` branch

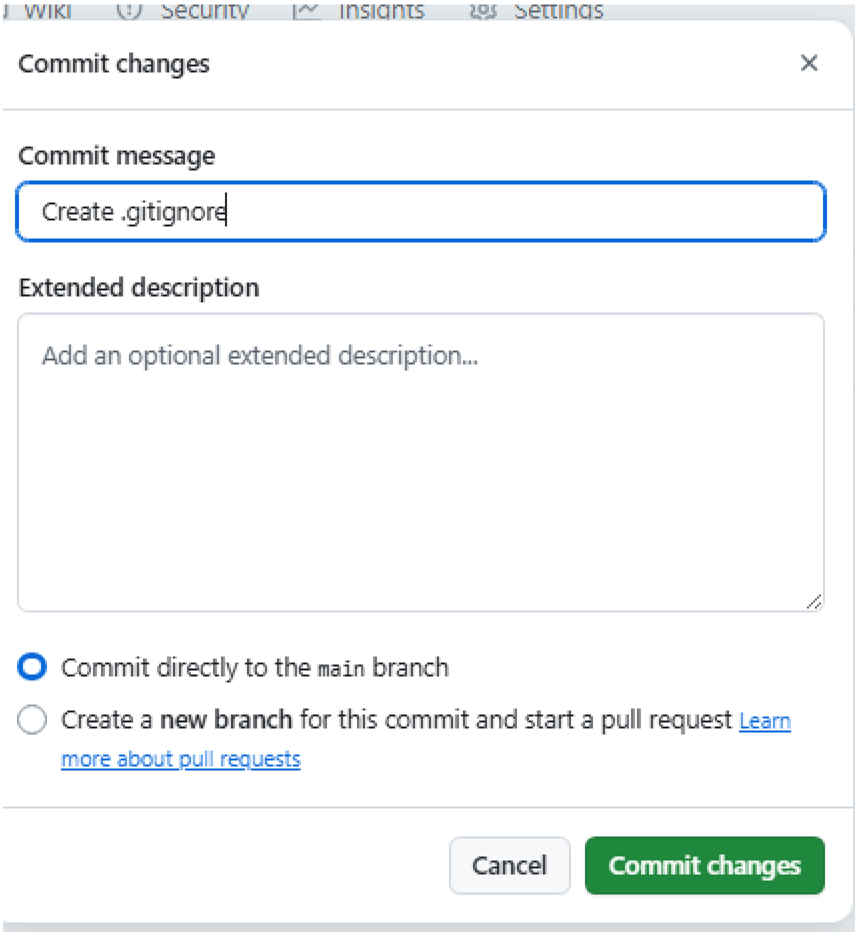
○ Create a **new branch** for this commit and start a pull request Learn more about pull requests

Add the .gitignore file and enter the details as follows and click on Commit changes


terraform_jenkins_pipeline / .gitignore in main
Cancel changes
Commit changes...
Edit
Preview
Choose .gitignore template: None
Spaces
2
No wrap

```
.terraform/
*.tfstate
*.tfstate.backup
*.log
.tfplan
```

Enter the Commit message and click on Commit changes



Commit changes

Commit message

Create .gitignore

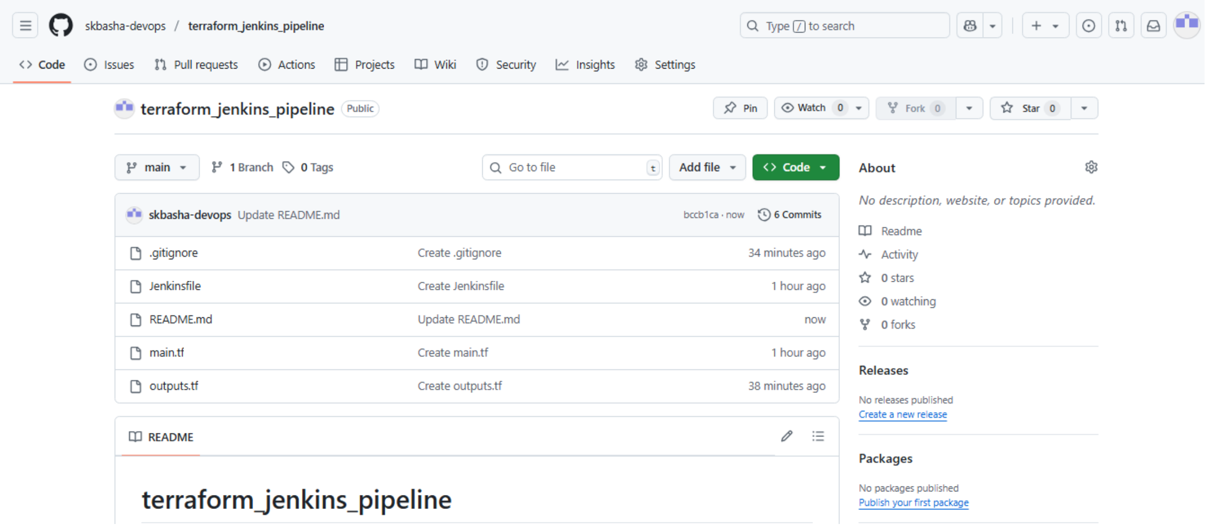
Extended description

Add an optional extended description...

● Commit directly to the main branch

○ Create a **new branch** for this commit and start a pull request Learn more about pull requests

All the files required are added



skbasha-devops / terraform_jenkins_pipeline

terraform_jenkins_pipeline Public

skbasha-devops Update README.md bccb1ca · now 6 Commits

.gitignore	Create .gitignore	34 minutes ago
Jenkinsfile	Create Jenkinsfile	1 hour ago
README.md	Update README.md	now
main.tf	Create main.tf	1 hour ago
outputs.tf	Create outputs.tf	38 minutes ago

README

terraform_jenkins_pipeline

About

No description, website, or topics provided.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Final Repo Structure:

terraform-jenkins-pipeline/

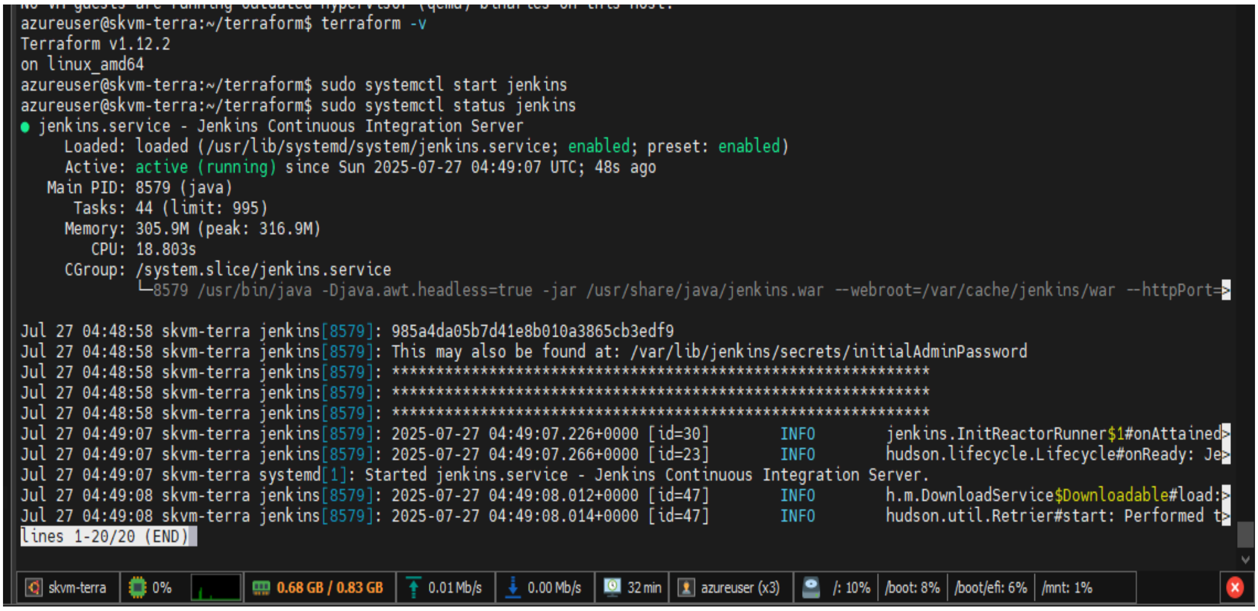
```
├── .gitignore (optional)
├── Jenkinsfile
├── README.md (optional)
├── main.tf
└── outputs.tf (optional)
```

Connect Jenkins to GitHub Repo

In the linux CLI check if terraform is installed and start Jenkins

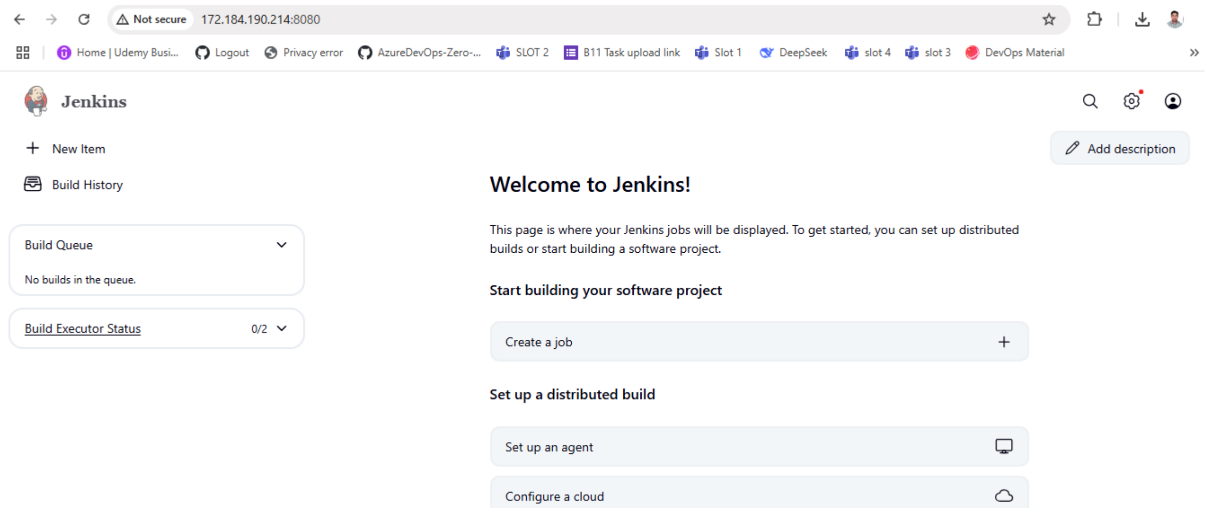
Run commands terraform --version

sudo systemctl start jenkins
sudo systemctl status jenkins



```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
azureuser@skvm-terra:~/terraform$ terraform -v
Terraform v1.12.2
on linux_amd64
azureuser@skvm-terra:~/terraform$ sudo systemctl start jenkins
azureuser@skvm-terra:~/terraform$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-07-27 04:49:07 UTC; 48s ago
   Main PID: 8579 (java)
      Tasks: 44 (limit: 995)
     Memory: 305.9M (peak: 316.9M)
        CPU: 18.803s
     CGroup: /system.slice/jenkins.service
             └─8579 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=>

Jul 27 04:48:58 skvm-terra jenkins[8579]: 985a4da05b7d41e8b010a3865cb3edf9
Jul 27 04:48:58 skvm-terra jenkins[8579]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Jul 27 04:48:58 skvm-terra jenkins[8579]: *************************************************************
Jul 27 04:48:58 skvm-terra jenkins[8579]: *************************************************************
Jul 27 04:48:58 skvm-terra jenkins[8579]: *************************************************************
Jul 27 04:49:07 skvm-terra jenkins[8579]: 2025-07-27 04:49:07.226+0000 [id=30]       INFO       jenkins.InitReactorRunner$1#onAttained>
Jul 27 04:49:07 skvm-terra jenkins[8579]: 2025-07-27 04:49:07.266+0000 [id=23]       INFO       hudson.lifecycle.Lifecycle#onReady: Je>
Jul 27 04:49:07 skvm-terra systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Jul 27 04:49:08 skvm-terra jenkins[8579]: 2025-07-27 04:49:08.012+0000 [id=47]       INFO       h.m.DownloadService$Downloadable#load:>
Jul 27 04:49:08 skvm-terra jenkins[8579]: 2025-07-27 04:49:08.014+0000 [id=47]       INFO       hudson.util.Retrier#start: Performed t>
lines 1-20/20 (END)
```

Open Google Chrome and enter 172.184.190.214:8080 , Sign-in to Jenkins using Username and Password



Build Queue

No builds in the queue.

Build Executor Status 0/2

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

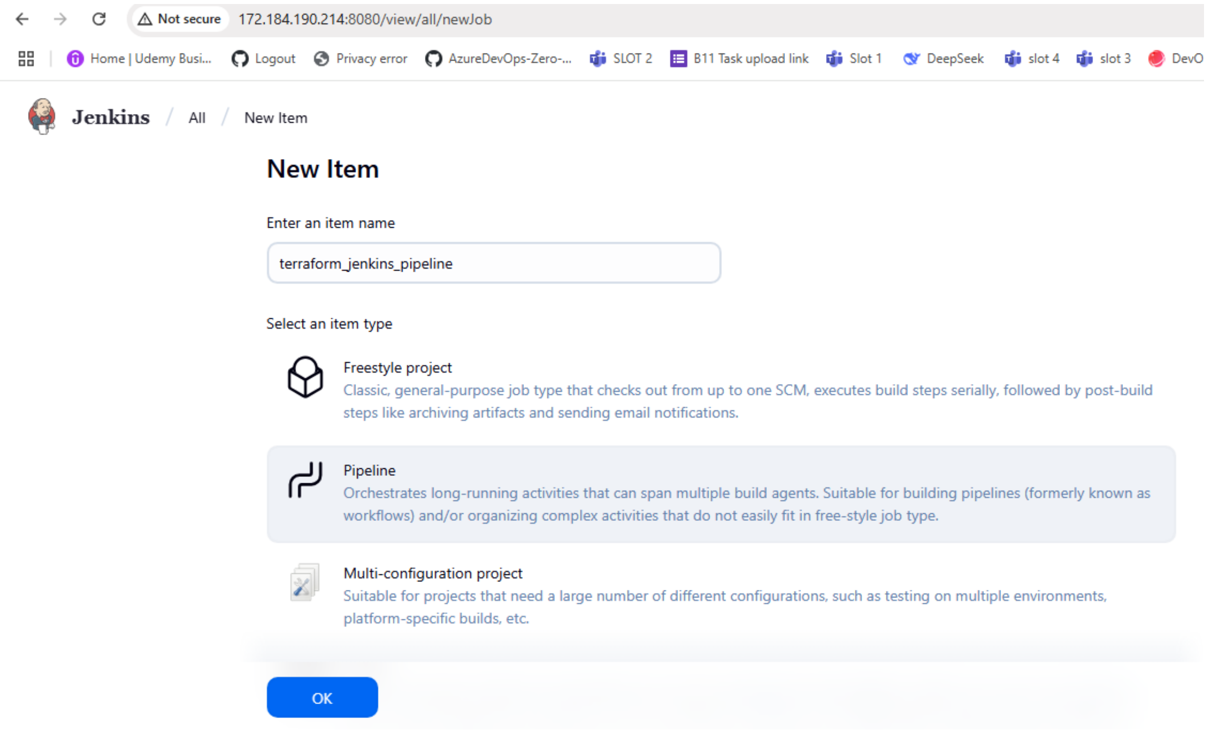
Create a job

Set up a distributed build

Set up an agent

Configure a cloud

In Jenkins Dashboard click on New Item and select Pipeline job and click on OK



New Item

Enter an item name

terraform_jenkins_pipeline

Select an item type

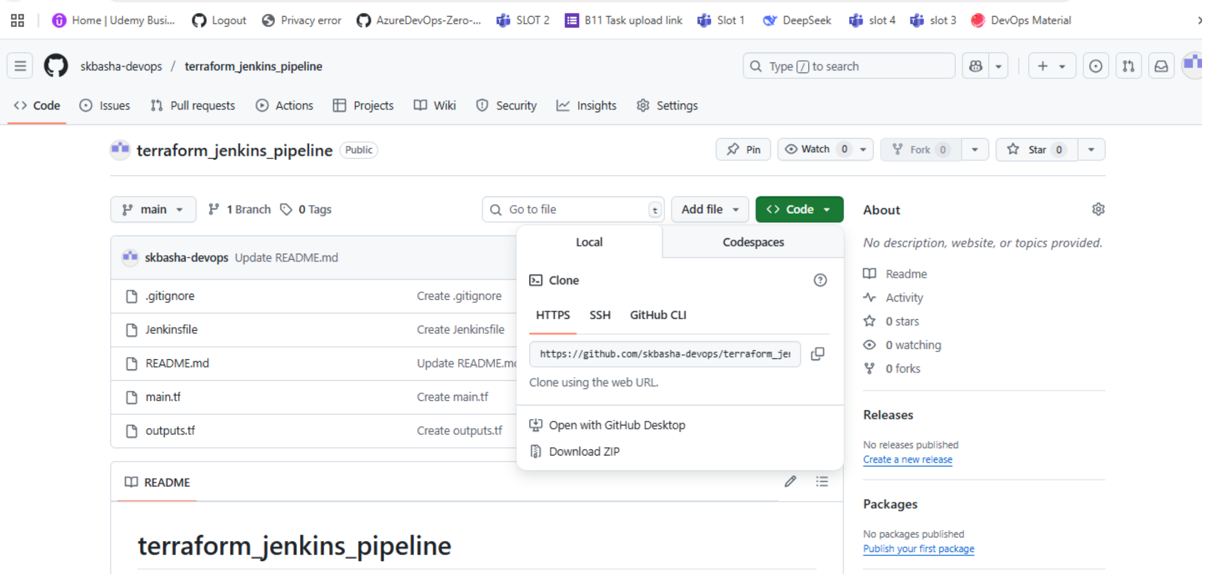
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

Copy the Repository URL



skbasha-devops Update README.md

.gitignore	Create .gitignore
Jenkinsfile	Create Jenkinsfile
README.md	Update README.md
main.tf	Create main.tf
outputs.tf	Create outputs.tf

https://github.com/skbasha-devops/terraform_je

Clone using the web URL.

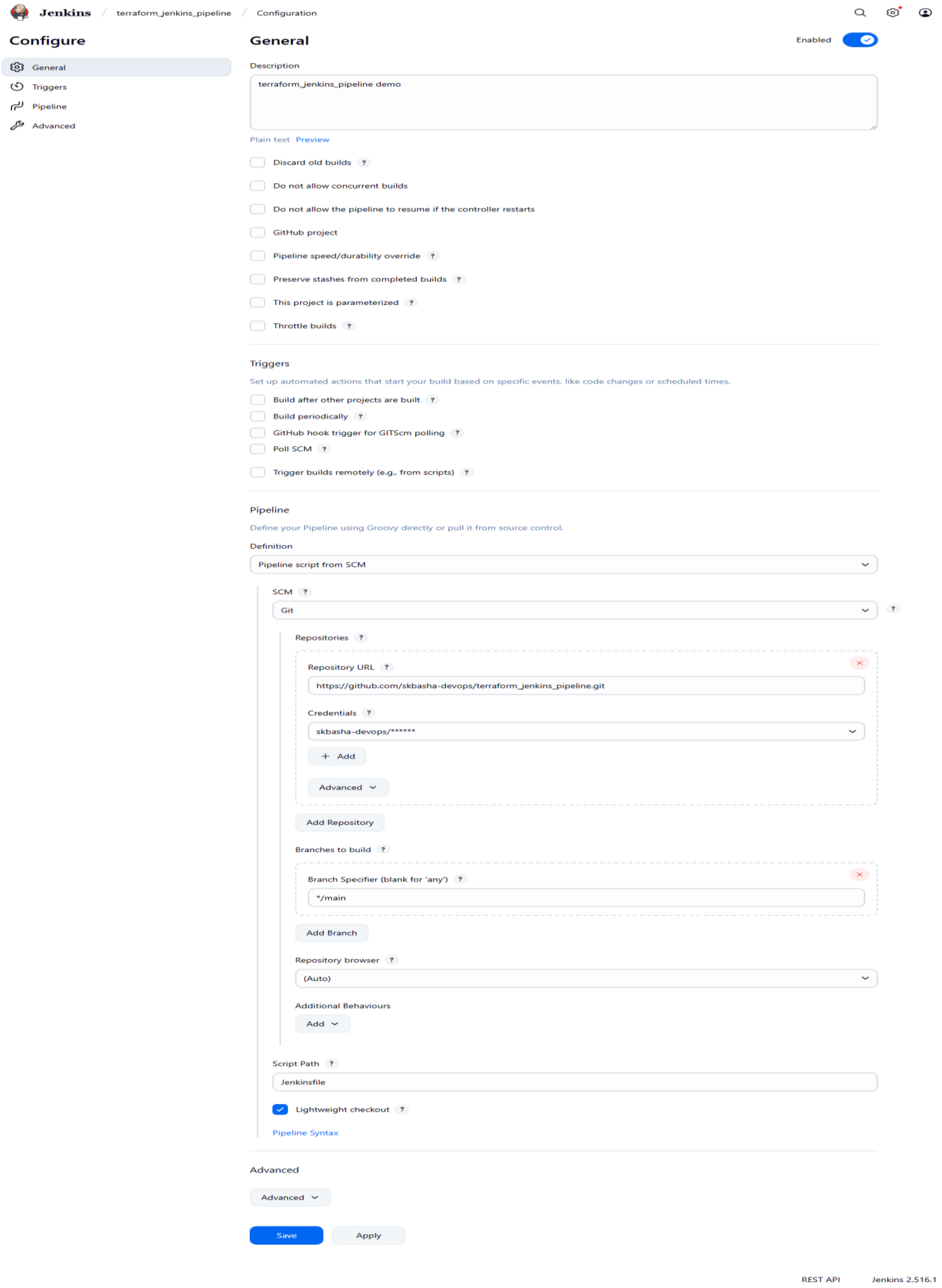
No description, website, or topics provided.

No releases published

No packages published

terraform_jenkins_pipeline

Enter the details as follows
Give description.
Use "Pipeline script from SCM", Paste the Repository URL, Select the credentials and give branch specifier as */main and click on Save



Jenkins / terraform_jenkins_pipeline / Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

General — Enabled ●

Description

terraform_jenkins_pipeline demo

Plain text Preview

□ Discard old builds ?
□ Do not allow concurrent builds
□ Do not allow the pipeline to resume if the controller restarts
□ GitHub project
□ Pipeline speed/durability override ?
□ Preserve stashes from completed builds ?
□ This project is parameterized ?
□ Throttle builds ?

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

□ Build after other projects are built ?
□ Build periodically ?
□ GitHub hook trigger for GITScm polling ?
□ Poll SCM ?
□ Trigger builds remotely (e.g., from scripts) ?

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

SCM ?: Git

Repositories ?

Repository URL ?: https://github.com/skbasha-devops/terraform_jenkins_pipeline.git

Credentials ?: skbasha-devops/******

+ Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?: */main

Add Branch

Repository browser ?: (Auto)

Additional Behaviours

Add

Script Path ?: Jenkinsfile

☑ Lightweight checkout ?

Pipeline Syntax

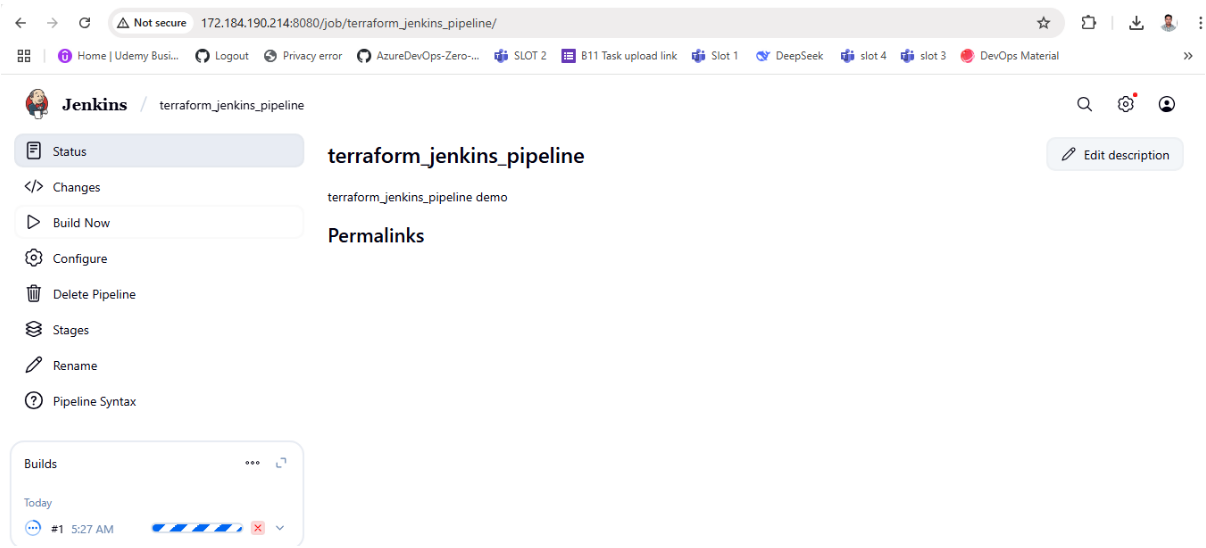
Advanced

Advanced

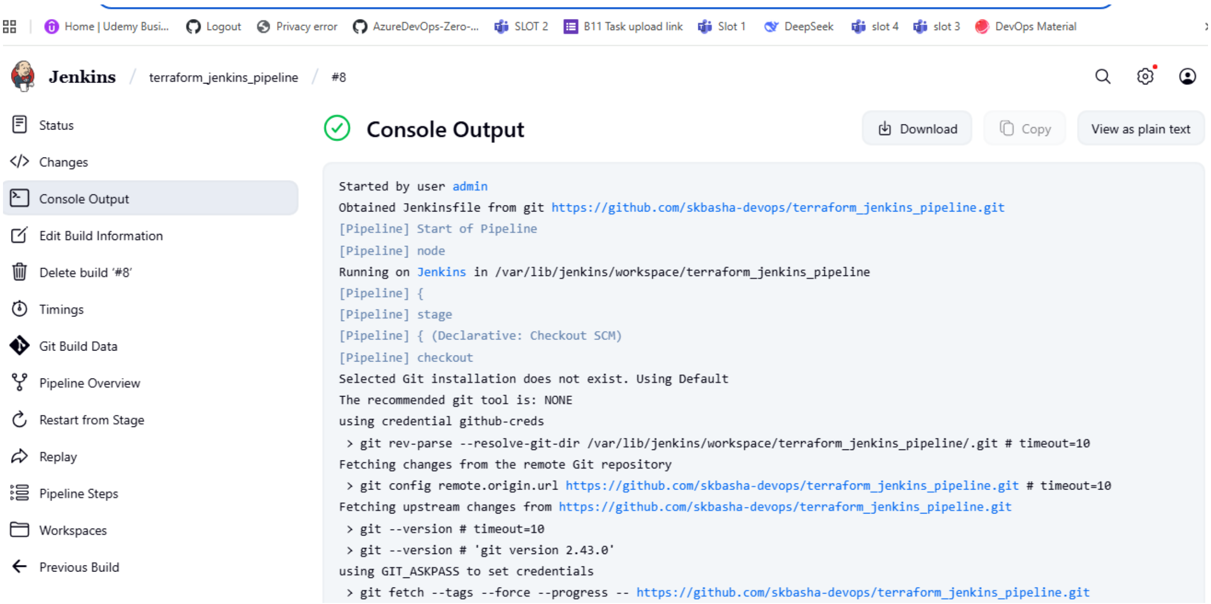
Save Apply

REST API Jenkins 2.516.1

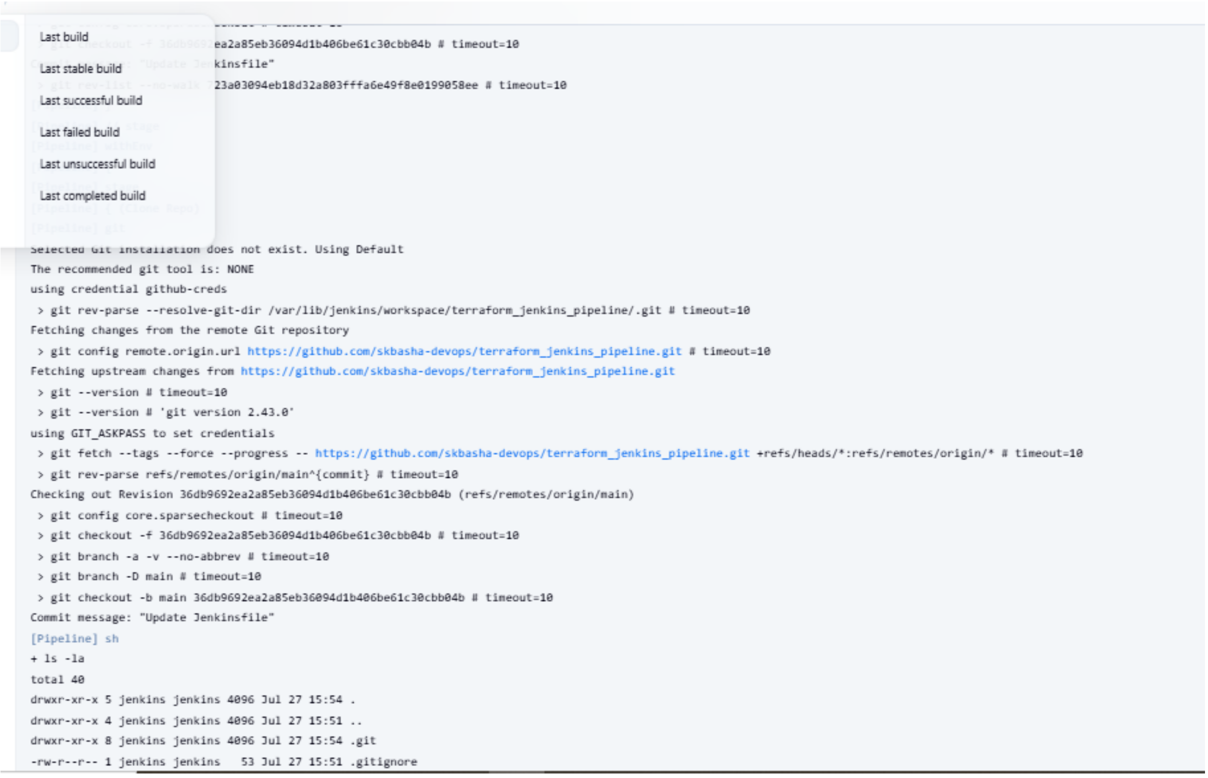
Click on Build Now



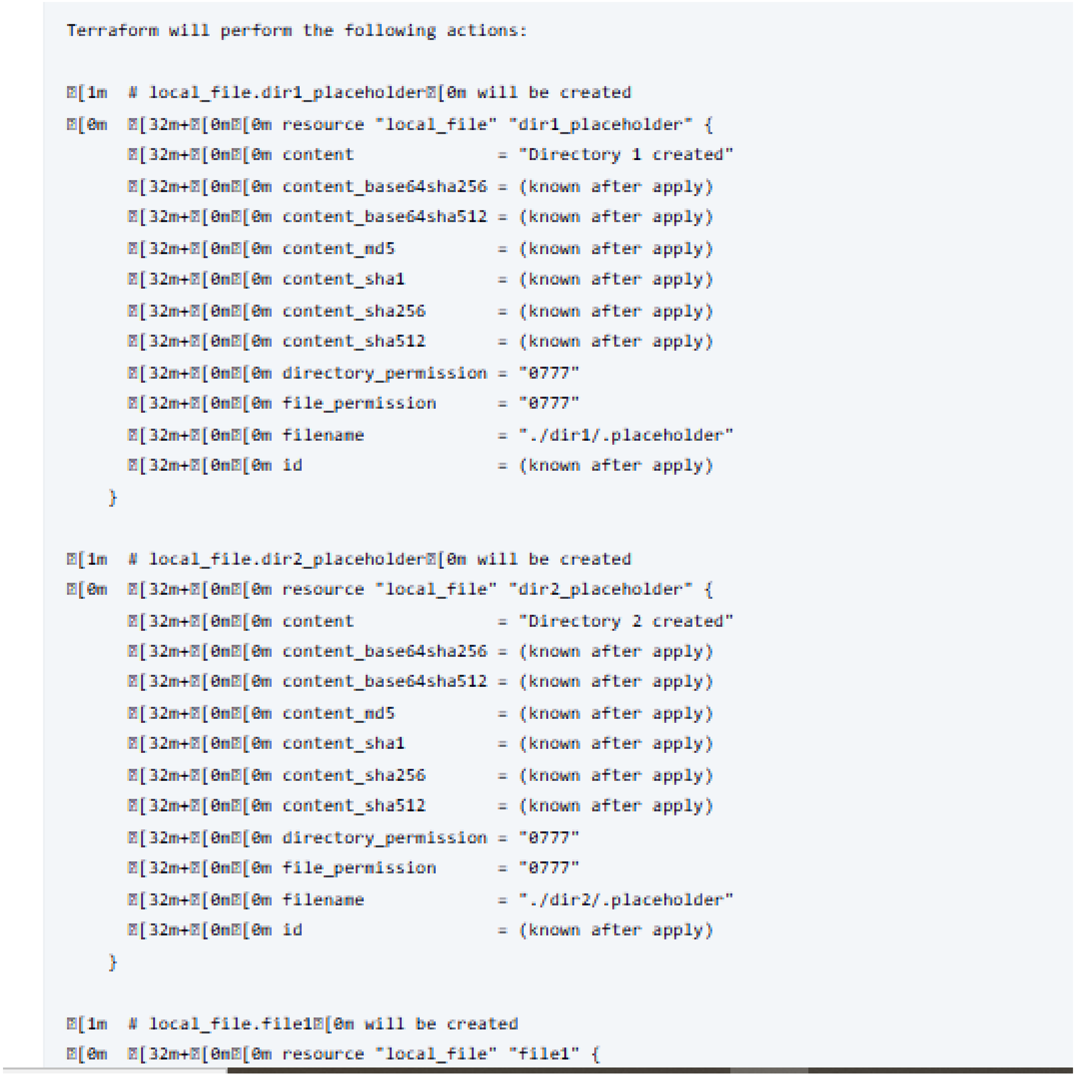
Build is successful, click on #8



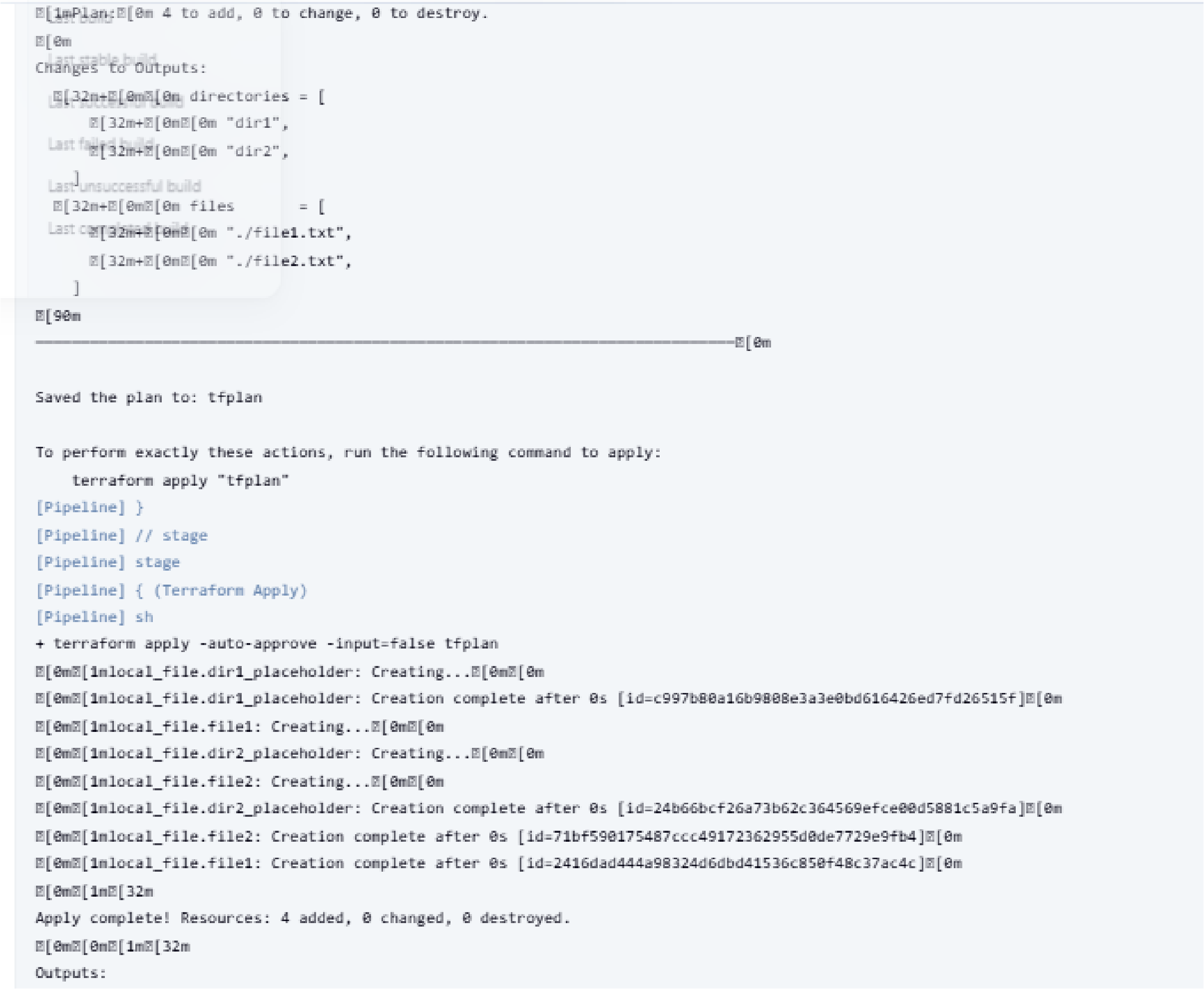
```
Started by user admin
Obtained Jenkinsfile from git https://github.com/skbasha-devops/terraform_jenkins_pipeline.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/terraform_jenkins_pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-creds
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/terraform_jenkins_pipeline/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/skbasha-devops/terraform_jenkins_pipeline.git # timeout=10
Fetching upstream changes from https://github.com/skbasha-devops/terraform_jenkins_pipeline.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress -- https://github.com/skbasha-devops/terraform_jenkins_pipeline.git
```



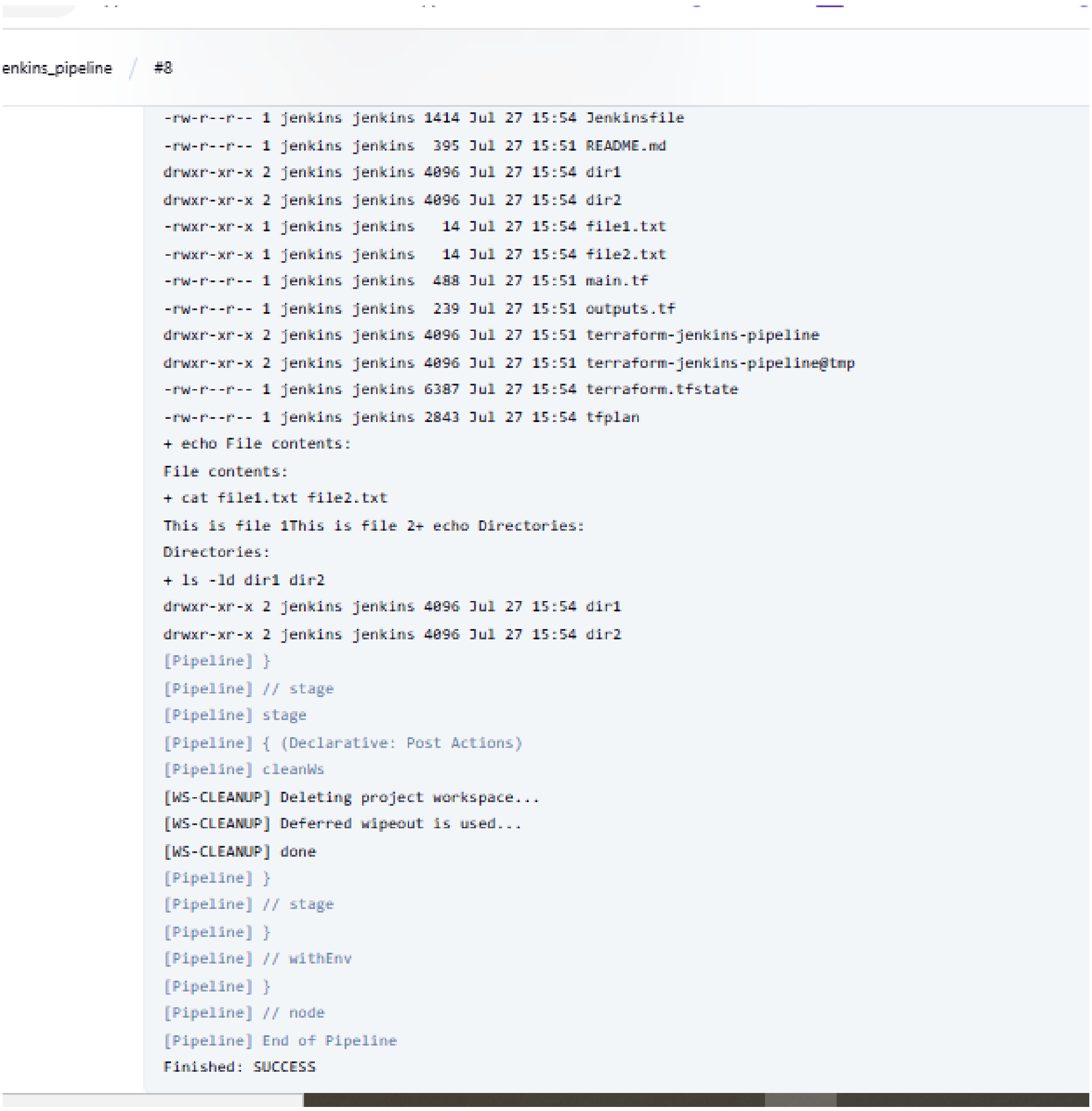
```
 > git checkout -f 36db9692ea2a85eb36094d1b406be61c30cbb04b # timeout=10
Commit message: "Update Jenkinsfile"
 > git rev-list --no-walk 723a03094eb18d32a803fffa6e49f8e0199058ee # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone Repo)
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-creds
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/terraform_jenkins_pipeline/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/skbasha-devops/terraform_jenkins_pipeline.git # timeout=10
Fetching upstream changes from https://github.com/skbasha-devops/terraform_jenkins_pipeline.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress -- https://github.com/skbasha-devops/terraform_jenkins_pipeline.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 36db9692ea2a85eb36094d1b406be61c30cbb04b (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 36db9692ea2a85eb36094d1b406be61c30cbb04b # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D main # timeout=10
 > git checkout -b main 36db9692ea2a85eb36094d1b406be61c30cbb04b # timeout=10
Commit message: "Update Jenkinsfile"
[Pipeline] sh
+ ls -la
total 40
drwxr-xr-x 5 jenkins jenkins 4096 Jul 27 15:54 .
drwxr-xr-x 4 jenkins jenkins 4096 Jul 27 15:51 ..
drwxr-xr-x 8 jenkins jenkins 4096 Jul 27 15:54 .git
-rw-r--r-- 1 jenkins jenkins   53 Jul 27 15:51 .gitignore
```



```
Terraform will perform the following actions:

⌧[1m  # local_file.dir1_placeholder⌧[0m will be created
⌧[0m  ⌧[32m+⌧[0m⌧[0m resource "local_file" "dir1_placeholder" {
      ⌧[32m+⌧[0m⌧[0m content              = "Directory 1 created"
      ⌧[32m+⌧[0m⌧[0m content_base64sha256 = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_base64sha512 = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_md5          = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_sha1         = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_sha256       = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_sha512       = (known after apply)
      ⌧[32m+⌧[0m⌧[0m directory_permission = "0777"
      ⌧[32m+⌧[0m⌧[0m file_permission      = "0777"
      ⌧[32m+⌧[0m⌧[0m filename             = "./dir1/.placeholder"
      ⌧[32m+⌧[0m⌧[0m id                   = (known after apply)
    }

⌧[1m  # local_file.dir2_placeholder⌧[0m will be created
⌧[0m  ⌧[32m+⌧[0m⌧[0m resource "local_file" "dir2_placeholder" {
      ⌧[32m+⌧[0m⌧[0m content              = "Directory 2 created"
      ⌧[32m+⌧[0m⌧[0m content_base64sha256 = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_base64sha512 = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_md5          = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_sha1         = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_sha256       = (known after apply)
      ⌧[32m+⌧[0m⌧[0m content_sha512       = (known after apply)
      ⌧[32m+⌧[0m⌧[0m directory_permission = "0777"
      ⌧[32m+⌧[0m⌧[0m file_permission      = "0777"
      ⌧[32m+⌧[0m⌧[0m filename             = "./dir2/.placeholder"
      ⌧[32m+⌧[0m⌧[0m id                   = (known after apply)
    }

⌧[1m  # local_file.file1⌧[0m will be created
⌧[0m  ⌧[32m+⌧[0m⌧[0m resource "local_file" "file1" {
```

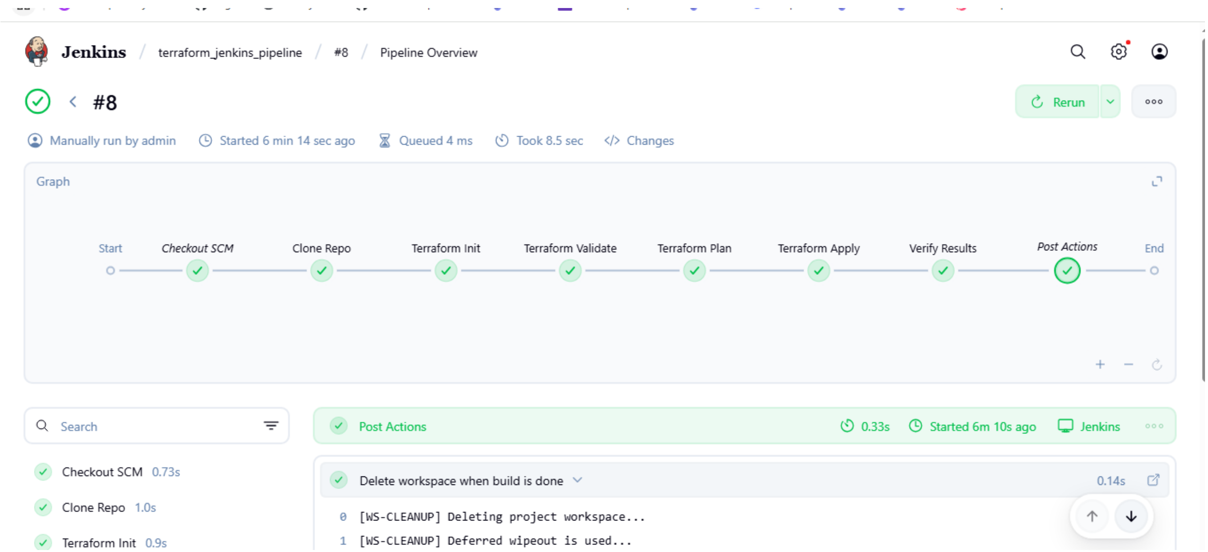


```
␛[1mPlan:␛[0m 4 to add, 0 to change, 0 to destroy.
␛[0m
Changes to Outputs:
  ␛[32m+␛[0m␛[0m directories = [
      ␛[32m+␛[0m␛[0m "dir1",
      ␛[32m+␛[0m␛[0m "dir2",
    ]
  ␛[32m+␛[0m␛[0m files       = [
      ␛[32m+␛[0m␛[0m "./file1.txt",
      ␛[32m+␛[0m␛[0m "./file2.txt",
    ]
␛[90m
─────────────────────────────────────────────────────────────────────────────␛[0m

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
    terraform apply "tfplan"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Terraform Apply)
[Pipeline] sh
+ terraform apply -auto-approve -input=false tfplan
␛[0m␛[1mlocal_file.dir1_placeholder: Creating...␛[0m␛[0m
␛[0m␛[1mlocal_file.dir1_placeholder: Creation complete after 0s [id=c997b80a16b9808e3a3e0bd616426ed7fd26515f]␛[0m
␛[0m␛[1mlocal_file.file1: Creating...␛[0m␛[0m
␛[0m␛[1mlocal_file.dir2_placeholder: Creating...␛[0m␛[0m
␛[0m␛[1mlocal_file.file2: Creating...␛[0m␛[0m
␛[0m␛[1mlocal_file.dir2_placeholder: Creation complete after 0s [id=24b66bcf26a73b62c364569efce00d5881c5a9fa]␛[0m
␛[0m␛[1mlocal_file.file2: Creation complete after 0s [id=71bf590175487ccc49172362955d0de7729e9fb4]␛[0m
␛[0m␛[1mlocal_file.file1: Creation complete after 0s [id=2416dad444a98324d6dbd41536c850f48c37ac4c]␛[0m
␛[0m␛[1m␛[32m
Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
␛[0m␛[0m␛[1m␛[32m
Outputs:
```



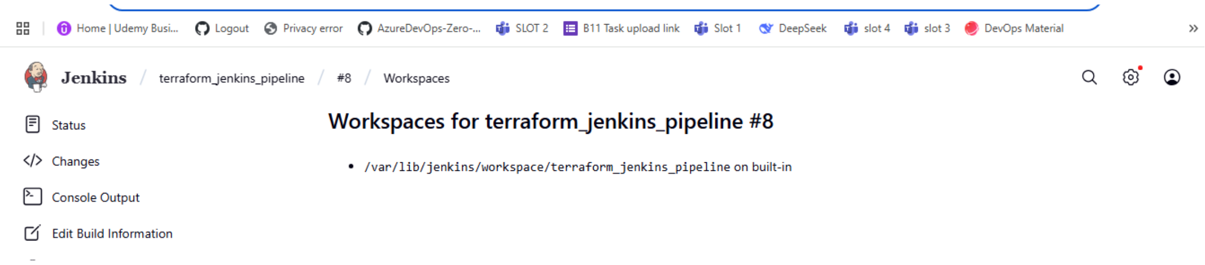
```
-rw-r--r-- 1 jenkins jenkins 1414 Jul 27 15:54 Jenkinsfile
-rw-r--r-- 1 jenkins jenkins  395 Jul 27 15:51 README.md
drwxr-xr-x 2 jenkins jenkins 4096 Jul 27 15:54 dir1
drwxr-xr-x 2 jenkins jenkins 4096 Jul 27 15:54 dir2
-rwxr-xr-x 1 jenkins jenkins   14 Jul 27 15:54 file1.txt
-rwxr-xr-x 1 jenkins jenkins   14 Jul 27 15:54 file2.txt
-rw-r--r-- 1 jenkins jenkins  488 Jul 27 15:51 main.tf
-rw-r--r-- 1 jenkins jenkins  239 Jul 27 15:51 outputs.tf
drwxr-xr-x 2 jenkins jenkins 4096 Jul 27 15:51 terraform-jenkins-pipeline
drwxr-xr-x 2 jenkins jenkins 4096 Jul 27 15:51 terraform-jenkins-pipeline@tmp
-rw-r--r-- 1 jenkins jenkins 6387 Jul 27 15:54 terraform.tfstate
-rw-r--r-- 1 jenkins jenkins 2843 Jul 27 15:54 tfplan
+ echo File contents:
File contents:
+ cat file1.txt file2.txt
This is file 1This is file 2+ echo Directories:
Directories:
+ ls -ld dir1 dir2
drwxr-xr-x 2 jenkins jenkins 4096 Jul 27 15:54 dir1
drwxr-xr-x 2 jenkins jenkins 4096 Jul 27 15:54 dir2
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] cleanWs
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Click on pipeline overview , here we can see the flow



```
0 [WS-CLEANUP] Deleting project workspace...
1 [WS-CLEANUP] Deferred wipeout is used...
```

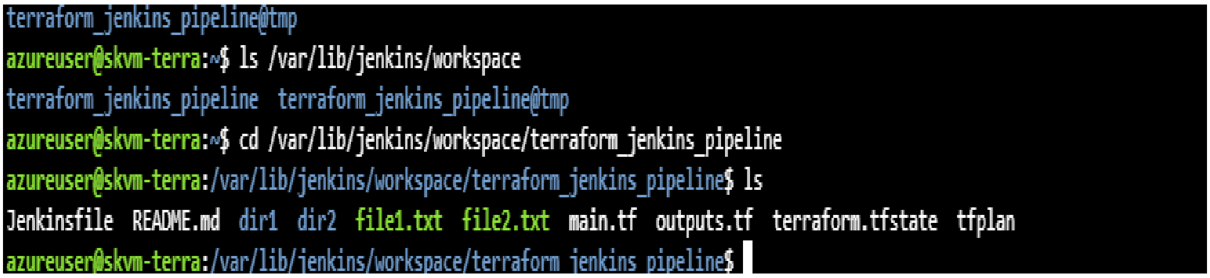
Click on Workspaces where the file are created



Workspaces for terraform_jenkins_pipeline #8

- /var/lib/jenkins/workspace/terraform_jenkins_pipeline on built-in

In CLI check if the files and directories are added



```
terraform_jenkins_pipeline@tmp
azureuser@skvm-terra:~$ ls /var/lib/jenkins/workspace
terraform_jenkins_pipeline  terraform_jenkins_pipeline@tmp
azureuser@skvm-terra:~$ cd /var/lib/jenkins/workspace/terraform_jenkins_pipeline
azureuser@skvm-terra:/var/lib/jenkins/workspace/terraform_jenkins_pipeline$ ls
Jenkinsfile  README.md  dir1  dir2  file1.txt  file2.txt  main.tf  outputs.tf  terraform.tfstate  tfplan
azureuser@skvm-terra:/var/lib/jenkins/workspace/terraform_jenkins_pipeline$
```